

Jack Dohany  
325 O'Connor St.  
Menlo Park, CA 94025

March 1986  
(415) 321-7684

\*\*\* INTRODUCTION TO CUSTOMIZED MSCRIPT VERSION 5 \*\*\*

Hi there! This sheet is not your only documentation. The rest is on the disc or wafer (or on Side B of the cassette) in the form of 3 MSCRIPT textfiles called ANN, BILL and CARL. These files form one reference work, with topics in alphabetical order. At your leisure you may load the files, edit them as you like to suit your printer, and print them. Or you may order a letter-quality hardcopy for \$8.00.

There are 4 varieties of CMS V5, named as follows:

MS5T for cassette only	MS5D for cassette and AERCO disc
MS5A for cassette and A&J	MS5Z for cassette and ZEBRA disc

In each case the BASIC module, when loaded, autostarts at line 900 and loads the machine code module. Both modules have the same file name. You will then see the HOME MENU, summarized on Page 2.

IMPORTANT STUFF:

1. A flashing "?" prompt indicates we are waiting for you to press a key, thus selecting a command or option, or answering a question. You may also BREAK; to resume operation, please RUN. Note that even when a catalog is on the screen, you may still press a key corresponding to any HOME MENU command or option.
2. You always go to MSCRIPT with Command T. When you have ADDED, LOADED or RESTORED text, you must then press ENTER to see your text. You should then DELETE the "ENTER" just written!
3. The MSCRIPT HELP MENU does not exist in CMS V5. It's been replaced with the cataloger which I think you'll find more useful. Function-H now takes you to the HOME MENU.
4. When loading, adding, killing or restoring text, keep an eye on "text length" at the bottom of the HOME MENU: it's the main indication that your wish has been fulfilled.
5. Stock MSCRIPT's tape handler is accessed from the MSCRIPT COMMAND MENU. It is NOT COMPATIBLE with HOME MENU tape commands other than CATALOG.
6. Whenever you get an IO ERROR HALT (such as DISC FULL or TAPE ERROR), just RUN to get going again. Don't panic, no harm done.
7. For Tasman type "B" interface, POKE 35879, 191.  
For Tasman type "C" interface, POKE 35879, 251.
8. Suggested PRINT MENU parameters for 2040 printer:  
PS=1,LM=0,LL=32. Home Menu: EJECT=N.

\*\*\* SUMMARY OF THE HOME MENU \*\*\*

ADD TEXT: Adds text from MSD to end of current text in RAM.  
BACKUP: Makes a backup of CMS on cassette, disc or wafer.  
CATALOG: Catalogs a cassette, disc or wafer.  
DRIVE: Selects another drive.  
ERASE: Permanently erases text file from disc.  
FORMAT: Formats a diskette or wafer.

INTERFACE: Allows selection of printer interface.  
KILL TEXT: Erases current text, in effect, but not really.  
LOAD TEXT: Loads a textfile from tape/disc/wafer.  
NUMCOPIES: Allows selection of number of copies to be printed.  
PAPER: Selects next possible paper/ink combination.  
QUIT: Stops program and "cleans" computer. RUN to resume.

RESTORE: Resurrects killed text, as much as possible.  
SAVE: Saves all or part of text to cassette/disc/wafer.  
TO MSCRIPT: Starts Mscript, preserving any existing text.  
USE: Selects cassette or disc/wafer.  
VIEW MEM: Goes to a peek/poke utility. Press M to exit.

X=BLOCK: If N, MS will print the entire text.  
          If Y, MS will print text as follows:  
              If no block markers, entire text is printed.  
              If 1 BLM, text is printed from that BLM to end.  
              If 2 BLMs, text is printed from 1st to 2nd BLM.

Y=EJECT: If Y, MS will do a form-feed at end of printing.  
          If N, printhead will halt at last text character.

Z=LFEED: If Y, MS will send a linefeed after each CR.  
          If N, printer must supply LF after Carriage Return.

NOTE: The CASSETTE variety lacks commands USE, DRIVE, ERASE and FORMAT. The A&J variety lacks the ERASE command.

\*\*\* NEW CMS FUNCTIONS IN MSCRIPT \*\*\*

Function-A: Turns ALTERNATE CHARACTER SET on/off.  
          (First sign of change: numbers at bottom of screen)  
Function-H: Goes to Home Menu.  
Function-K: Turns KEYCLICK on/off.  
Function-X: Flips top keyrow.  
          (Cursor=X when FLIPPED, and Cursor=L/C when NORMAL)  
Function-Z =BREAK = SYMBOL-SHIFT+Break key

\*\*\* THE TOP KEYROW NOW LOOKS LIKE THIS \*\*\*

1	2	3	4	5	6	7	8	9	0
ISP	CAP	TAB	FLF	LFT	DWN	UP	RT	DLF	DEL

ISP means INSERT SPACE at CURSOR.  
FLF means CURSOR FULL LEFT; again, FULL RIGHT.  
DLF means DELETE CHARACTER AT LEFT OF CURSOR.  
DEL means DELETE CHARACTER AT CURSOR.

\*\*\* CUSTOMIZED MSCRIP \*\*\*  
Version 5

NOTE 1: Topics discussed are in alphabetical order so you can find them fast. A row of asterisks heads each new topic. Line numbers refer to BASIC lines where the command or option is processed. LOC refers to memory locations. All numbers are in good old decimal.

NOTE 2: There are 4 varieties of Customized Mscript Version 5:

MSST for cassette only  
MSSA for cassette and A&J wafer  
MSSD for cassette and AERCO disc  
MSSZ for cassette and ZEBRA disc

This document discusses all four varieties. I'm too lazy to produce 4 separate documents.

NOTE 3: CMS MC means Customized Mscript Machine Code. Machine code buffs may purchase a source listing of CMS MC as an MSCRIP text on cassette, disc or paper for \$10. The topic titled MACHINE CODE explains briefly the function of MC numbers found in the BASIC listing.

NOTE 4: If you like CMS, please send me MONEY. The only money I get for the enormous work of developing CMS comes from the good people who use it. I suggest at least \$8. What you pay a dealer is kept by the dealer for materials, handling and mailing.

NOTE 5: Customized Mscript remains COPYRIGHTED by Micro Systems Inc. I have provided backup facilities with CMS for your convenience as a legal MSCRIP owner. Please do not abuse these backup facilities.

NOTE 6: Thanks are offered to the many people who have helped with this project or offered suggestions, especially: David Lockin, Mark Fisher, Tom Bent, P. H. Skipper, Doug Dewey, Tom Woods, Norm Lenfeldt, Dick Scoville, and Pat Morrissey. And thanks to Steve Pagliarulo and Micro Systems for bringing MSCRIP to the 2068 in the first place.

\*\*\*\*\*  
ADD TEXT: (Line 100) Allows you to add text from disc or tape onto the tail end of text in memory. Added text will be truncated if it's too big, EXCEPT for AERCO! Aenco users must ensure that text to be added will fit! Otherwise, ADD TEXT works just like LOAD TEXT, which please see. Note that after ADDING, LOADING or RESTORING text, you must press ENTER once, after going to MSCRIP with Command T.

The "current document name" as reported on the Home and MSCRIP Command menus is updated only after LOADING or SAVING a whole document, and not after adding a document or saving a block.

\*\*\*\*\*  
ALT CHARS (FUNCTION-A): Allows the Alternate Character Set to be turned on or off; they are initially on. The Alternate Character Set is used by the screen and by the 2040 printer, but not by other printers. The alternate character set's vertical lines are fatter than normal, which makes them easier to read on most monitors and TVs. When you press FUNCTION-A the only immediate change will be in the numbers at the bottom of the screen. Pressing ENTER will cause the whole screen to change. Status of FUNCTION-A is preserved in backups.

\*\*\*\*\*  
BACKUP (Line 110): Allows a backup copy of MSS to be made on tape or disc/wafer. You must give a NAME.

When doing a backup on ZEBRA disc, please do not include a type with the name. You may use the name "start" if you want the program to auto-load.

With AERCO disc, you can use the name BOOT if you want the program to autoloan.

Always do a backup on a DIFFERENT cassette, disc or wafer than the current one.

\*\*\*\*\*  
BLOCK: The segment of text between the first two block markers. There may be more than two BM's in text and they may not all be visible on the screen. So, if you are going to mark a block you should form the habit of first using FUNCTION-U to remove any unseen block markers, and then use FUNCTION-B to insert your block markers.

\*\*\*\*\*  
BLOCK OPTION (Line 110) may be Y or N. Always "N" in stock mscript. I should have called this BLOCK-PRINT MODE, but there wasn't enough room on menu.

If N, MSCRIP always prints the whole text.  
If Y, MSCRIP prints text according to these rules:  
IF no Block Markers, print whole text.

IF 1 BM, print text from that BM to end.  
IF 2 or more BMs, print from BM1 to BM2.

Note that printing a block is just like printing a whole text. When printing a block, MSCRIP doesn't even look at the rest of text and so does not see any format lines that may be there.

\*\*\*\*\*  
BLOCK MARKER: a code 14 in text. Inserted in text at cursor by FUNCTION-B. All block markers may be removed from text with FUNCTION-U. Block operations affect only text from BM1 to BM2.

The block marker symbol on screen is a vertical line with Fat Characters off, and a special symbol with Alternate Characters on: the upper 2/3 of a "plus" sign. I think it's easier to see.

\*\*\*\*\*  
BLOCK SAVE: If there is a marked block in your text, the Home Menu SAVE command gives you the option of saving just that block as a mini-textfile. Only the text between the block markers is saved, and not the block markers themselves.  
A 0 (end-of-text marker) is added at the end of the block, by substituting a 0 for the block marker in the text and later replacing it.

Some neat uses for BLOCK SAVE: say you are writing a very large document in several parts, since it won't all fit in memory at once, and you want to move a paragraph from one part to another. Easy, just save the paragraph you want to relocate, and later append it to the other text when you have it in memory.

Or say you're typing away and all of a sudden you run out of memory. Easy, just mark the first half of your text, go to the home menu and save the block, go back to MSCRIP, remove the block and continue working.

Then there's "boilerplate", which is just a hunk of text that you might need to use often and don't want to retype every time, such as a paragraph explaining that the check really is in the mail, or that CMS will be done real soon, please be patient. Or maybe it's a particular set of printcode definitions that you use regularly!

\*\*\*\*\*  
BLOCK COPY: =FUNCTION-C. Will DUPLICATE and insert a marked block of text at the cursor, without removing the original marked block. Can repeat by holding FUNCTION-C down.

\*\*\*\*\*  
BLOCK REMOVE: =FUNCTION-R. Permanently deletes a marked block of text. Use with caution! first use BLOCK UNMARK, then mark the block you wish to copy or remove. Then copy it if you want. Then remove it.

\*\*\*\*\*  
BLOCK UNMARK: =FUNCTION-U. Removes any and all block markers from text, including any you can't see on the screen. Should always be used FIRST, before marking a block.

\*\*\*\*\*  
BOILERPLATE: see BLOCK SAVE

\*\*\*\*\*  
BOLDFACE: type FUNCTION-G then + to turn BOLDFACE PRINTING on if off, or off if on. NOTE that this does NOT cause any printer-specific control codes to be sent to the printer. Rather, MSCRIP simply sends a BACKSPACE after each character and sends the character again, causing it to be printed twice. This makes for slow printing. Your printer may very well have several better ways of printing boldface using imbedded printcodes. Note that some printers (ink-jets) do not have a BACKSPACE function, and thus MSCRIP's + and - printcodes will NOT work with those printers.

Note that MSCRIP's + and - printcodes are intended mainly for use with "stupid" printers, such as typewriters converted for use as printers, which would not otherwise have boldface or underlining capability.

\*\*\*\*\*  
BREAK may be used freely when in BASIC whenever a flashing M or V prompt is showing. ALWAYS RUN when you are ready to restart; this will harm nothing. Also see QUIT. Note that BREAK does NOT leave the computer in a condition to load other programs, whereas QUIT does.

\*\*\*\*\*  
BREAK: The MSCRIP BREAK function has been REDEFINED by CMS MC as SYMBOL-SHIFT+BREAK rather than CAPS-SHIFT+BREAK. This is because the BREAK key and the SPACE bar are hardwired together on the 2068, and pressing SPACE while holding down CAPS-SHIFT should NOT produce a BREAK, but merely a space.

Note that MSCRIP BREAK may be used to interrupt printing unless CMS option EJECT is N. MSCRIP BREAK is also used to get from text to MSCRIP Command Menu and to return to text from any MSCRIP menu.

\*\*\*\*\*  
CATALOG (Line 120): Allows you to find out just what's on a tape, disc or wafer. This command works a little differently for each kind of media, so each will be discussed separately. The USE option must be set to the appropriate medium.

\*\*\*\*\*  
CATALOG for Cassette, AERCO and A&J allows headers to be sent (in

groups of e) to the screen and (optionally) to MSCRIPT TEXT for subsequent editing and printing. Headers sent to text are added to the end of the current text. You might like to add tape counter numbers to such a text for cassette, or other comments about a particular file.

A header is a short file (17 bytes) that prefaces the main file and contains all sorts of information about the file. The cataloger sends a title line for every six headers.

In the title line, TYP means TYPE OF FILE, and there are these possible types:

BAS: a BASIC program  
 DAT: a numeric data array  
 CHR: a character array  
 BIN: a binary or CODE file, or text file  
 MTX: a stock MSCRIPT cassette textfile

With AERCO DISC these types are also possible:

SCR: a screen file  
 ARD: an ARDS-type file  
 LRD: an LROS-type file  
 VAR: all BASIC variables  
 BUT: a BOOT file

Also in the title line, NAME means the filename which can vary in length for different media. START means the starting line number of an autostart BASIC program or the starting address of a CODE file. LENTH means the total length of a BASIC program including variables, or the length of a CODE file. PLEN means the length of a BASIC program excluding variables. Thus length of variables = LENTH-PLEN.

In the cases of DAT and CHR types, the letter-name of the array will appear in the column titled START.

If you plan to catalog to text, I suggest that you first save any current text and KILL the current text to provide lots of room for cataloging. The software will report "TEXT FULL" when that becomes true, and will continue sending headers to the screen only.

If you've elected to send headers to text, the screen will scroll without asking you. If not, it will ask the "scroll?" question and wait for your reply. Just press Y (=ENTER) or N, but not BREAK or SPACE, which would interrupt the program.

\*\*\*\*\*  
 CATALOG for ZEBRA does NOT read disc headers, nor does it catalog to text, simply because I can't get enough information about the TDS to figure out how to do it. TDS does however allow you to specify a "DIR TYPE" (up to 3 letters, no dot); it will then pluck out information just on files of that type and send them to

the screen. If you just hit ENTER when asked for "DIR TYPE", the entire disc catalog will be sent to the screen without stopping. I don't know of any way to make it pause, despite what the TDS manual says.

\*\*\*\*\*  
 CATALOG for CASSETTE works alike on all 4 varieties. It's like a program called HEADERS, except that you have the option of sending the headers to text, AND the software WILL also handle Stock Mscript tape headers. The latter contain only a name and text length.

\*\*\*\*\*  
 CATALOG for AERCO is quite simple, it just sends the headers to the screen and/or text, exactly like the A&J and CASSETTE varieties, but of course it's faster. If you want to do a standard CAT "", command, you can of course QUIT and do it; the information provided by this command does not go to text.

\*\*\*\*\*  
 CATALOG for A&J works like this: first it moves the tape to the "End of Tape marker", and then it begins reading headers, and it COUNTS them. It saves the first header it finds and compares each subsequent header to the first one; a match is taken to mean that we have gone around the loop, and the current count (held in LOC 24202) is the number of the last file recorded.

The HOME MENU adds 1 to this number and displays it as the NEXT file number.

There is a problem with this system: IF there is a header on the wafer which is identical to the first header, then the cataloger will quit TOO SOON, and the NEXT FILE NUMBER will be quite incorrect, and you won't have cataloged the entire wafer. Normally this won't happen, unless you are in the unfortunate habit of making identical backups with identical names on the same wafer.

There IS a provision in the software for this eventuality: LOC 24203 normally contains 1. If you poke 0 into that location, the software will NOT match the first header with subsequent headers, and will go on reading headers from the wafer indefinitely, until you hit the SPACE bar. In this case also, the NEXT FILE NUMBER as reported by the Home Menu will be quite untrustworthy!

The reason for this difficulty is that it is apparently impossible to look for an End-Of-Tape marker while also looking for a header, without major revision to the A&J operating system, which is in EPROM and not easily modified.

\*\*\*\*\*  
 COMMENT LINE: Information or remarks that you want to include in

your text but which you don't want printed. Begins with ">" and ends with a carriage return. It may contain any remarks you like, which you don't want printed. Thus, if you want to "suppress" a paragraph, you can just insert ">" at the front of it.

\*\*\*\*\*  
 CURSOR: The CMS MSCRIPT cursor reflects the status of CAPS LOCK, on key 2. If the upper keyrow is normal (not flipped) then a flashing L means lower-case and a flashing C means caps-lock on. If the top keyrow is flipped the cursor will be a flashing X. See FLIP.

\*\*\*\*\*  
 CURSOR BLINK RATE is held in LOC 37116, and is normally 1. You can slow it down by making it 2,3,4, or 5. This does not affect keyboard response, but only how fast the cursor blinks. VIEW MEM can be used to change the rate.

\*\*\*\*\*  
 DELETE (key 0): deletes the character under the cursor.

DELETE LEFT (key 9): a new CMS function which deletes the character to the left of the cursor. Many word processors have both kinds of DELETE available, and now so does MSCRIPT.

\*\*\*\*\*  
 DRIVE (line 130) (AERCO, A&J and ZEBRA only): Allows you to select a different drive. Displays current drive. ZEBRA ONLY: when first loaded, current drive is assumed to be A, which may or may not be correct.

\*\*\*\*\*  
 DISC COMMANDS: CMS has the most-used disc commands built in. Of course you can QUIT or BREAK and then use other disc commands, for example to copy from one drive to another, or (ZEBRA only) protect or unprotect a particular file. There is just not enough room in BASIC to provide all possible disc commands on the Home Menu.

\*\*\*\*\*  
 EJECT OPTION (Line 140): If Y (as in stock MSCRIPT), paper is always advanced to the head of the next page when printing is done. If N the printer halts right after printing is done. Useful for the 2040, and for testing, and maybe for other things. CAUTION: a quirk is that when EJECT is N the program does NOT test to see whether the break key is being pressed or whether the printer is ready, but continues to try printing. I can't find a way around this quirk. Maybe you can.

\*\*\*\*\*  
 ERASE (line 140) (AERCO and ZEBRA only): Allows you to ERASE any unprotected disc file. It does NOT erase text from memory. You are given the chance to change your mind. NAME (and, with ZEBRA, TYPE) must be correctly given.

\*\*\*\*\*  
 FIND (FUNCTION-F): Will move the cursor to the start of a string which has been previously defined from the MSCRIPT Command Menu. Will repeat if keys held down.

\*\*\*\*\*  
 FLIP: FUNCTION-X. This is a CMS function which allows you to "flip" the upper keyrow so you can use cursor controls WITHOUT pressing CAPS-SHIFT. But note that when the top keyrow is in a flipped condition you MUST use caps-shift to type NUMBERS. The top keyrow will flip whenever you use FUNCTION-X. The cursor will reflect the status of the upper keyrow: C/L cursor means normal, X means flipped.

Most computers have separate keys for cursor controls. The 2068 is not so endowed: FLIP is an attempt to compensate for that lack. You will love it or hate it...do at least try it for a while. The reasoning behind FLIP is that most writers use the cursor controls far more frequently than they type numbers. To use FLIP effectively one MUST form the habit of using CAPS SHIFT when typing numbers. The problem is that one must then shuck that habit when using the 2068 for other things, which is kind of like switching from a car with automatic transmission to one with manual.

\*\*\*\*\*  
 FORMAT (Line 150) (AERCO, A&J and ZEBRA only): Allows you to format a new disc or wafer, or totally erase an old one. The disc or wafer to be formatted must be unprotected.

A&J's formatting is quite straightforward, and has no provision for naming a wafer.

AERCO gives the disc a random serial number (based on the 3 system variables called FRAMES) and asks for an optional name.

ZEBRA, however, requires a name and will abort the format operation if no name is given. ZEBRA also requires that a good formatted disc be on drive A at the start of formatting. I

suggest that the formatted disc you use should be protected. The disc that you want to format should be on drive B; but if you have only one drive, you will have to remove the formatted disc and insert the disc to be formatted when asked to do so.

\*\*\*\*\*

FORMAT LINE: A text line that will not be printed and which contains COMMANDS to be used by MSCRIPT during printing subsequent text. Also see COMMENT LINE.

A Format Line must begin with a ">" in column 1. A Format Line is terminated by a carriage return. Commands in a format line are separated by a comma.

Commands are actually definitions and have the form "cc=x" where cc means the 2-character name of what is being defined, and "x" is either a number, text or Y or N.

Text may include \$ for which MSCRIPT will substitute the current page number. And it may include 1 or more slashes (/) as "positioners" for the text. See pg. 37 of manual.

One thing that can be defined is a PRINTCODE, thus: #0=27/0. See IMBEDDED PRINTCODES.

Another thing that can be defined is anything on the PRINT MENU. Thus >L=12,LL=68\ would redefine the left margin as 12 and the line length as 68.

LIST OF FORMAT LINE COMMANDS: p31 means Page 31 in manual.

JU=Y/N	JUSTIFY LEFT	p31	FN=nbbr	PAGE NUMBER	p40
CE=Y/N	CENTER	p31	TT=text	TOP TITLE	p37
FR=Y/N	FLUSH RIGHT	p33	ET=text	EVEN-PN TT	p37
HI=nbbr	HANGING INDENT	p33	OT=text	ODD-PN TT	p37
SS=Y/N	SINGLE SHEET	p27	BT=text	BOTTOM TITLE	p38
LM=nbbr	LEFT MARGIN	p31	EB=text	EVEN-PN BT	p38
LL=nbbr	LINE LENGTH	p31	OB=text	ODD-PN BT	p38
LS=nbbr	LINE SPACING	p27	HM=nbbr	HEADER MARGIN	p39
FL=nbbr	PAGE LENGTH	p27	FN=nbbr	FOOTER MARGIN	p39
FS=nbbr	PAGE SPACING	p27			

\*\*\*\*\*

FUNCTION refers to any MSCRIPT command that can be used while you are writing text. A FUNCTION is used by holding down both CAPS SHIFT and SYMBOL SHIFT with two fingers on your right hand and pressing a numeric or letter key with whatever spare finger you may have available. However, some functions are on the top keyrow and need only CAPS SHIFT held down (unless the top keyrow is flipped).

LIST OF MSCRIPT FUNCTIONS: (\*\*CMS only; p19=Page 19 of manual)

#0=27/0, #1=52/0, #2=53/0

">" must be in column 1 and means "this is a format line".  
 "#0" means "this is a printcode definition".  
 "0" is the printcode token itself.  
 "27" is a decimal code, in this case an ESC which tells a printer that 1 or more control codes follow.

"/0" tells MSCRIPT that the printcode as you intend to use it is NOT a printable character. "/1" would mean that it IS a printable character. You could for example define a printcode as "129" which on many printers IS a printable character; MSCRIPT cannot know what kind of printer you have. MSCRIPT needs this information for keeping track of how many characters appear on a page-line.

The tokens "+" and "-" are predefined as "boldface on/off" and "underlining on/off". Most (but not all) printers will recognize the codes sent when these tokens are imbedded. Also see BOLDFACE and UNDERLINING.

You might also like to know that a code 15 is used to represent the copyright symbol (FUNCTION-G) in text. When MSCRIPT sees a code 15 it knows that the next character should be a token. If it's not, it will be ignored.

\*\*\*\*\*

INITIAL PRINT PARAMETERS: When MSCRIPT starts up it uses the contents of certain memory locations to initialize the print parameters that you see on the print menu. You can change what you see on the menu, but if you regularly use different parameters than normal, you can use VIEW MEMORY to change the initial print parameters, and do a backup. Then whenever you load that backup, your parameters will already be the way you regularly use them. A list of locations and what they normally contain (shown in parentheses) follows:

36952 (64)	WINDOW	36968 (63)	LINE LENGTH
36956 (60)	PAGE LENGTH	36972 (1)	LINE SPACING
36960 (6)	PAGE SPACING	36974 (1)	PAGE NUMBER
36964 (8)	LEFT MARGIN		

Use care not to change the contents of any nearby locations.

If you only own a 2040 printer, you would want LM to be always 0 and LL to be 32, and PS to be 1 or 2, to avoid wasting paper.

\*\*\*\*\*

INSERT BLOCK: =FUNCTION-I. This function has NOTHING WHATEVER to do with a marked BLOCK of text, and would be better called "INSERT MODE". To use it, simply move your cursor to where you want to insert some text, press FUNCTION-I, and type away. To

S: FULL LEFT	I: INSERT-MODE
G: DOWN A SCREEN	K: KEYCLICK ON/OFF
U: UP A SCREEN	M: MEND (exit INSERT MODE)
T: TAB	N: NEW PAGE
A: ALT CHARS ON/OFF	P: PRINT MENU
B: BLOCK MARKER	Q: QUIT CHANGES = UNDO
C: COPY BLOCK	R: REMOVE BLOCK
D: DELETE	S: SUB-DELETE
E: TO END OF TEXT	T: TO TOP OF TEXT
F: FIND STRING	U: UNMARK BLOCK
G: IMBED TOKEN	X: FLIP TOP KEYROW
H: HOME (BASIC) MENU	Z: COMMAND MENU

See KEYBOARD CHANGES for a list of upper-keyrow functions.

\*\*\*\*\*

HELP (FUNCTION-H) has been redefined by CMS as HOME, because I needed the memory taken up by the MSCRIPT HELP menu, which was simply an incomplete list of MSCRIPT functions.

HOME (FUNCTION-H) jumps to the HOME (or BASIC CMS) menu. (You can also go HOME from the MSCRIPT COMMAND menu key R.)

\*\*\*\*\*

IMBED A PRINTCODE: (FUNCTION-G) This function allows you to imbed a SINGLE TOKEN which can be a "+", a "-", or a digit from "0" to "9". A token "0" to "9" is called a PRINTCODE and always represents a SINGLE NUMBER from 0 to 255. You should only imbed printcodes which you have defined earlier in the text, using one or more format lines.

NOTE: a PRINTCODE token does NOT represent a whole series of numbers as with TASHWORD's imbedded graphic characters. To send the printer a series of numbers you must imbed a series of tokens. Each token you use MUST have been defined in text prior to being imbedded. MSCRIPT initializes the value of all PRINTCODE tokens as the number 0 which, if sent to a printer, will do nothing whatever.

Assuming that PRINTCODE 0 were defined as 27 and PRINTCODE 1 as 52 and PRINTCODE 2 as 53, the following would print the word "time" in italics on an Epson printer:

Now is the \$0\$itimes\$0\$2 for all good men and women to come...

NOTE that the \$ stands for FUNCTION-G and would show up on your screen as a copyright symbol.

With Tasword, graphic characters are defined in the BASIC program, because Tasword lacks MSCRIPT's Format Line system. With MSCRIPT, Printcodes can and must be defined using format lines. An example:

get out of INSERT MODE, use FUNCTION-M, which I call MEND. Using DELETE or ENTER or moving the cursor vertically will also exit Insert-mode. NOTE that you can automatically enter INSERT MODE simply by placing your cursor to the right of a carriage-return symbol and typing away!

\*\*\*\*\*

INTERFACE OPTION (Line 180): allows you to select one of 4 printer interfaces. ALSO: for Tasman "B", LOC 35879 must contain 191; for Tasman "C", LOC 35879 must contain 251. See and use VIEW MEMORY.

\*\*\*\*\*

INSERT SPACE, a new CMS function on key 1, causes a space to be inserted at the cursor. VERY handy, I use it a lot. It can be used to position words on a screen line.

\*\*\*\*\*

KEYCLICK DURATION can be adjusted by poking new values into LOC 35462 (small effect) and LOC 35463 (big effect).

KEYCLICK PITCH can be adjusted by poking new values into LOC 35464 (small effect) and LOC 35465 (big effect).

You can harm nothing by poking these locations. Note that BASIC uses the same keyclick as MSCRIPT, except during INPUTS.

\*\*\*\*\*

KEY REPEAT DELAY can be adjusted by poking a new value into LOC 42515. This controls how much time elapses when you are holding down a key until it begins to repeat. People with manual impairment may wish to lengthen this. LOC 42515 normally contains 8. Increasing it to 16 would double the delay. This cannot be tested from BASIC.

\*\*\*\*\*

KEYBOARD CHANGES MADE BY CMS:

\*\* THE TOP KEYROW NOW LOOKS LIKE THIS \*\*

1	2	3	4	5	6	7	8	9	0
ISP	CAPS	TAB	FLP	LFT	DN	UP	RT	DLE	DEL
new		dup	dup					new	

"dup" means a stock function has been duplicated on this key.

"new" means stock MSCRIPT didn't have this function.

"FLP" means "CURSOR FULL LEFT".

"ISP" means "INSERT SPACE".

"DLE" means "DELETE character at LEFT of cursor".

"DEL" means "DELETE character at cursor". Also see FLIP.

\*\*\*\*\*

KEYBOARD ERRORS: It sometimes happens that a single keystroke

will produce TWO or more characters. This is because MSCRIPT's key handler routine is extremely sensitive, to ensure that a keystroke is never missed. Keystrokes are stored in a "queue" in the order received, and processed in that order.

However, there is "inadequate debouncing" for the 2068's chintzy keyboard, and I haven't yet found a way to provide better debouncing. So I recommend using the keyclick and watching out (listening out?) for extraneous keystrokes.

\*\*\*\*\*  
KILL TEXT (Line 200) will apparently ERASE any text in memory. The only evidence you will immediately see is that the text length as shown at the bottom of the screen will become 1. Text so erased can be recovered by using the RESTORE command. All the KILL TEXT command does is to put a zero (end-of-text mark) into LOC 46927, the start of the text file.

\*\*\*\*\*  
LFEEED OPTION (Line 350): If Y, a linefeed is sent to printer after every carriage return. If N, the printer must do its own linefeed after each carriage return; otherwise all printing will take place on one line. If your printer's auto-linefeed DIP switch is ON, the LFEEED option should be N. And vice-versa.

\*\*\*\*\*  
LOAD TEXT (Line 210): allows you to load text from tape (2068 style only) or AERCO disc or ZEBRA disc or A&J wafer, depending on status of USE option.

You do NOT need to use "@" with A&J, or "S" with ZEBRA; the software does that for you.

NAME must be same as when the text was saved. With ZEBRA, the name may include a type (a period followed by up to 3 letters).

If a name includes such a type, the type MUST be given as part of the name of the text you want to load.

ZEBRA and AERCO don't care about upper/lower case in a name, but cassette and A&J are quite fussy about it.

A name is optional with tape, but if you don't specify a name, you should be sure that the tape is positioned at the start of a text file. The loaded text will REPLACE the text in memory. However, see RESTORE.

Note that you CAN indeed load TASNWORD or other ASCII textfiles into MSCRIPT, though there will be some incompatibilities in how control codes are handled.

MSCRIPT texts are loaded, saved and stored in memory as CODE.

Therefore, when loading a text you should be sure that what you are loading is in fact a textfile and not a machine-code program. The CATALOG command can help; if you see that the starting address of a file is 46927, then it is most likely an MSCRIPT text. However, the ZEBRA catalog function does not give the starting address of a file (for some strange reason), but only its length. Using ZEBRA TYPES can help you here.

\*\*\*\*\*  
MACHINE CODE: The enhancements provided by CMS are mostly in machine code. When you LIST the BASIC portion of CMS you will see quite a few POKES, PEES and RANDOMIZE USRs, followed by a memory location or by a variable which has been defined as a memory location. Here is a list of those locations used by BASIC and a brief description of what they mean or do.  
System variables are discussed in 2068 manual, Pg 260-265.

27606,7 system variable CHARS  
27609 system variable PIP  
27692 system variable SCR CT  
27856 used to test whether AERCO board is connected.  
24201 A&J current drive; 0 or 1.  
24202 A&J current file # (from cataloger)  
24203 A&J cataloging mode; 1=seek match, 0=don't

-----CMS MC VARIABLES preserved in a backup-----

35440 Run-flag; used to determine how to start MSCRIPT.  
0 means cold-start, 1 means return, 2 means warm-start  
35441 USE: 0=cassette, 1=disc or wafer.  
35442 TOTEX: 1=catalog to text, 0=don't.  
35443 BCT: a counter used in sending 6 headers at a time.  
35444 FAT: 0=alt chars off, 1=on, used by MSCRIPT only.  
35445 PAPER: stores paper color  
35446 BLOCK-PRINT MODE: 0=off, 1=on  
35447 Number of copies to print  
35454,5 Text length  
35456,7 0, or loc of BM1  
35458,9 0, or loc of BM2  
35460 Printer I/F: 1=AER, 2=TAS, 3=AJ, 4=2040  
35461 Linefeed switch: 0=off, 1=on  
35462,3 DUR: keyclick duration  
35464,5 PIT: keyclick pitch

-----CMS MC ROUTINES CALLED BY BASIC-----  
(many others are patched into and accessed by MSCRIPT)

35567 SVO: sets Video Mode 0 (called at QUIT)  
35572 SVBO: sets video mode 128, leaving room for DF2.  
35616 TMS: preserves BASIC stack etc. and starts MSCRIPT.  
-looks at Run-Flag to see how to start MSCRIPT.  
3578a BEI: makes a keyclick, using DUR and PIT.  
3578c PUTNAM: moves n# into CURNAM.  
35772 SCROLL (used by View Mem)  
36779 CAT: the cataloging program for cassette/AJ/AERCO.  
43310 DISMEN: finds text length etc. and displays HOME MENU.  
43518 MENUS: the home menu itself, to LOC 43782.  
43627 CURDRV, part of home menu  
43783 UPDR: increments AERCO current drive in dock bank.  
44935 CURNAM: the current file name

\*\*\*\*\*  
MENUS: CMS V5 has 4 menus: the HOME MENU, the VIEW MEM MENU, MSCRIPT's COMMAND MENU, and MSCRIPT's PRINT MENU.

Stock MSCRIPT's HELP MENU has been eliminated in CMS V5, to gain room in memory for the cataloger. Sorry.

From the HOME MENU you may go to MSCRIPT text with key T or to the VIEW MEM menu with key V. You get back to the HOME menu from the VIEW MEM menu with key M, and from MSCRIPT TEXT with FUNCTION M.

MSCRIPT's COMMAND MENU is entered from MSCRIPT text by holding down SYMBOL SHIFT and pressing BREAK. FUNCTION-Z will also do it. You may go from the MSCRIPT COMMAND MENU to the HOME MENU by pressing key R and then ENTER.

MSCRIPT's PRINT MENU is entered with FUNCTION-P; see FUNCTION.

MSCRIPT's MENUS may both be EXITED by holding down SYMBOL SHIFT and tapping BREAK.

\*\*\*\*\*  
NCOPIES OPTION (Line 230) allows you to specify how many copies of a text you want printed. This is normally 1. If you make it 2 it will remain 2 until you change NCOPIES. For example, when I am writing letters I always set NCOPIES to 2: one copy for the recipient and one for me. NCOPIES can be as big as 100. If HEAD is off (N) there will be no form feeds between copies; this would be handy if you wanted to print up a bunch of "slips" containing just a paragraph or so. See HEAD.

\*\*\*\*\*  
PAPER (Line 250) allows you to cycle through the 8 possible paper/ink color combinations. In 2068 64-column mode, which MSCRIPT uses, only these combinations are possible, and BORDER color must always be same as paper.

\*\*\*\*\*  
PRINTCODES: See IMBEDDED PRINTCODES

\*\*\*\*\*  
QUIT (line 260) does MORE than just stop the program; it also sets the 2068 up so it is ready to load any other program. You may also stop by using BREAK (see BREAK), but if you then try to run another program, you invite disaster. CAUTION: whenever you use QUIT or BREAK, the ONLY way to restart is with RUN. This will NOT hurt anything you have already set up.

Use BREAK rather than QUIT when you want to find out how much room is left in BASIC with PRINT FREE.

\*\*\*\*\*

QUIT CHANGES = FUNCTION-Q = UNDO any deletion you have made using SUB-DELETE. Can restore only one screen line if used immediately. CANNOT restore a removed BLOCK! It's really gone. Will ALSO undo any INSERTION made while in INSERT MODE.

\*\*\*\*\*  
RESTORE (Line 270): Allows lost text to be recovered, if it was lost by use of the KILL or LOAD TEXT commands.

The Restore command works by replacing the end-of-text marker (0) with a carriage return (13).

\*\*\*\*\*  
SAVE TEXT (Line 280): Allows you to save the current text on cassette or disc/wafer. A NAME is required.

With ZEBRA, a TYPE may be given as part of the name, for example, "L3030786.LW". (This could mean letter #3 of date 3/7/86, to someone whose last name begins with W.)

With A&J you must also give a file #. This would normally be the number of the next unused file. (Do NOT use punctuation marks for file numbers greater than 9.) Thus, if the last file you put on a wafer was #11, then the next one should be #12. The cataloger can help you determine the next unused #. But it is NOT foolproof; see CATALOG A&J.

\*\*\*\*\*  
SAVING A BLOCK: If there are at least two BLOCK MARKERS in your text when you are saving it, you will be asked whether you wish to save JUST that block, rather than the whole text.

There can be a small problem with saving a block: before the block is saved, BASIC substitutes a 0 for the second block marker temporarily, so that the block will be saved as a "complete little textfile" including an end-of-text marker (a 0). After the save is done, the block marker is replaced.

BUT if there is an IO error halt for any reason (such as no more room on disc), and you then hit RUN (as indeed you always should after an IO error halt), then BASIC had no chance to replace the second block marker and so your text ends right there! -If this should happen, simply use the RESTORE command.

\*\*\*\*\*  
SUB-DELETE = FUNCTION-S. Allows you to delete all or part of a screen line. SPACE or ENTER provide escapes. Provided cursor has not been moved vertically, FUNCTION-Q will restore what was deleted.

\*\*\*\*\*  
TAPE: You may load or save text on tape in two different ways: by using MSCRIFT's own built-in tape handler, or by using the HOME

MENU Save/Load commands. NOTE that the two ways are NOT COMPATIBLE; that is, text saved to tape using the MSCRIFT command menu may NOT be loaded using the HOME MENU Load command. This is because MSCRIFT's built-in tape handler works differently from the standard 2048 BASIC tape handler.

\*\*\*\*\*  
TAB = FUNCTION-B; also duplicated on key 4 by CMS. Mscrift does not put TAB control codes into text as some word processors do. TAB in MSCRIFT is simply used for moving the cursor to predefined screen columns.

You can of course define and label TAB control codes in your text, but I see no good reason for doing so.

If you try to load an ASCII text produced by another word processor into MSCRIFT's text area (which you can certainly do), any tab control codes in that text will cause minor problems; you will see garbage characters on the screen. What you should do then is to use MSCRIFT to delete the garbage characters, and insert spaces as desired.

\*\*\*\*\*  
TAB TABLE: MSCRIFT's Command Menu allows you to define tabs as numbers from 1 to 132 corresponding to column positions. BUT there's a problem: the Command Menu only allows you to enter up to 16 CHARACTERS, and this can't be easily changed. So, the number of tabs you can define is severely limited, even though the TAB TABLE can actually hold 25 tabs!

You can get around this as follows: return to BASIC, poke the TAB TABLE with the stops you want, and go back to MSCRIFT. Do not load or append text before going back, or MSCRIFT will initialize the tab table.

The TAB TABLE is 26 bytes long, from LOC 44893 to 44918. It has these RULES:

1. It contains a list of up to 25 tabs.
2. A "tab" is a number ranging from 0 to 131, corresponding to column positions 1 to 132.
3. Tabs must be in sequence, as: 5 10 15 not 5 15 10.
4. The LIST of TABS in the table MUST be terminated by a 255 which signifies the end-of-table! Thus a complete list might be 5 10 15 20 25 255, which would mean tabs ONLY at columns 6,11,16,21 and 26. Thus your list of tabs can be as short or as long as you like, up to a maximum of 25 tabs; the 26th LOC would have to be a 255.

Using VIEW MEMORY, you can easily poke in any tab table you want.

\*\*\*\*\*  
TO MSCRIFT (Line 290): Goes to MSCRIFT, preserving any text in memory. If you have loaded or appended text, MSCRIFT will be necessarily restarted and you must immediately press ENTER to "capture" and display the textfile. In this case, your print parameters will also be re-initialized, as well as your tab settings, window etc.

\*\*\*\*\*  
UNDERLINING: There are two ways of underlining: 1) Imbedding a "-" token, and 2) imbedding special codes to use your particular printer's underline mode, if it has one. Some printers have several modes of underlining. Consult your printer manual to see what underlining modes it has, how they work, and what codes to define and imbed.

Imbedding a "-" turns MSCRIFT's built-in underlining mode on or off. In this mode, MSCRIFT does NOT send any printer-specific control codes to your printer. Rather, it simply sends a backspace and then a "-" after each character is sent. This is so that ANY printer which recognizes a backspace can be used to underline.

Note that the "-" character as printed by your printer may very well be less than a full character wide, in which case the underlining will appear broken.

Also see P. 32 of the stock MSCRIFT manual.

\*\*\*\*\*  
VIEW MEMORY (Line 310; line 500) is a special utility that allows you to PEEK and POKE around in memory. This allows you to tailor CMS somewhat to suit your needs and desires, but only following my directions, please. POKING the wrong place could have rather unpleasant consequences, so it would be well to SAVE anything precious before POKING.

The VIEW MEMORY menu is quite short:

ENTER increments the current location by one.

SPACE decrements the current location by one.

Note that ENTER and SPACE repeat, so you can scan forward or backward in memory.

MENU takes you back to the main menu.  
NEW LOC lets you input a new location.  
POKE takes you to POKE MODE.

A flashing ? prompt at the bottom screen indicates that any of the above commands may be given with a single keypress.

The CURRENT LOCATION is always shown at the bottom of the screen, followed by its contents as a decimal number and as a character (when possible).

In POKE MODE you can use any VIEW MEMORY menu command, but ENTER must also be pressed. You can also type in a number from 0 to 255 and press ENTER. That number will be POKED into the current location, and the current location will be incremented by one. This is for your convenience when poking a series of numbers, which would be the case if you were entering a TAB table.

POKE MODE may be exited by pressing E and then ENTER.

== END ==

